Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-31. (Canceled)

32. (Currently amended): A method of transforming data, the method comprising:

positioning a definition pointer to point at a first compound transform definition within a transform process definition;

invoking a first parallel processing thread to read the pointed at first compound transform definition;

searching data to be transformed for a data element to be transformed, the search being responsive to the first compound transform definition;

calling a dynamic function defined in the first compound transform definition, the dynamic function located elsewhere from the definition pointer position;

transforming any found data element into an output data file, responsive to the first compound transform definition and called dynamic function, a data structure of the output data file being responsive to a data structure of the transform process definition;

positioning a definition pointer to point at a second compound transform definition within the transform process definition;

invoking a second parallel processing thread to read the pointed at second compound transform definition;

searching data to be transformed for another data element to be transformed, the search being responsive to the second compound transform definition; and

transforming any found data element into the output data file, responsive to the second compound transform definition, the data structure of the output data file being responsive to the data structure of the transform process definition~~.~~

~~wherein the read first compound transform definition includes a translation codeset parameter enabling the transforming to include a call to one of a function or a lookup table located in the first compound transform definition~~.

33. (Previously presented): The method of claim 32, further including determining a type of the read first compound transform definition and, if the first compound transform definition is not a simple transform definition type, recursively calling the method of claim 32.

34. (Original): The method of claim 32, further including determining if all sub-definitions of a compound transform definition have been processed.

35. (Original): The method of claim 32, wherein the method of transforming data includes nesting of a data element.

36. (Previously presented): The method of claim 32, further including, if no data element is found in either step of searching data to be transformed, adding an output data element to the output data file responsive to the read first compound transform definition, the data to be transformed having no contribution to the output data element.

37. (Previously presented): The method of claim 32, wherein the read first compound transform definition includes a value parameter configured to specify a value for inclusion in the output data file.

38. (Original): The method of claim 32, wherein the data element is a compound data element and the read transform definition includes a source record parameter configured to specify the compound data element.

39. (Previously presented): The method of claim 32, wherein the read first compound transform definition is in a meta-language format.

40. (Original): The method of claim 32, wherein the data to be transformed data is in a meta-language data format.

41. (Previously presented): The method of claim 32, wherein the read first compound transform definition includes a transform element having an output field name and a source field parameter.

42. (Previously presented): The method of claim 32, wherein the read first compound transform definition includes a value parameter configured to populate a field in the output data file.

43. (Canceled)

44. (Currently amended): A method of transforming data, the method comprising:

positioning a definition pointer to point at a first compound transform definition, the first compound transform definition being within a transform process definition;

invoking a first parallel processing thread to read the pointed at first compound transform definition and sub-definitions of the first compound transform definition;

positioning a first payload pointer to point at a data element to be transformed, the positioning being responsive to a data structure of the first compound transform definition;

calling a dynamic function defined in the first compound transform definition, the dynamic function located elsewhere from the definition pointer position;

transforming the data element into an output data file, responsive to the read first compound transform definition and called dynamic function;

positioning the definition pointer to point at a second compound transform definition, the second compound transform definition being within the transform process definition;

invoking a second parallel processing thread to read the pointed at second compound transform definition and sub-definitions of the second compound transform definition;

positioning a second payload pointer to point at a data element to be transformed, the positioning being responsive to a data structure of the second compound transform definition; and

transforming the second data element into the output data file, responsive to the read second compound transform definition.

45. (Previously presented): The method of claim 44, further including determining a type of the read first compound transform definition and, if the read first compound transform definition is not a simple transform definition type, recursively calling the method of claim 44.

46. (Previously presented): The method of claim 44, further including determining a type of the read first compound transform definition and, if the read first compound transform definition is not a simple transform definition type, recursively calling the method of claim 44, wherein the recursive call is responsive to the data structure of the transform process definition.

47. (Previously presented): The method of claim 44, further including determining a type of the read first compound transform definition, if the read transform definition is not a simple transform definition type recursively calling the method of claim 44, and determining if all sub-elements of a compound element have been transformed.

48. (Original): The method of claim 44, further including determining if all sub-elements of a compound element have been transformed and, if the determination returns a value of YES, returning to a calling process.

49. (Canceled)

50. (Original): The method of claim 44, wherein the method of transforming data includes un-nesting of the data element to be transformed.

51. (Previously presented): The method of claim 44, wherein the read first compound transform definition includes a source field parameter configured to specify the data element.

52. (Previously presented): The method of claim 44, wherein the read first compound transform definition includes a source record parameter configured to specify the compound data element.

53-54. (Canceled)

55. (Original): The method of claim 44, further including a step of combining the data element with the transform process definition prior to transforming the data element to output data.

56. (Original): The method of claim 44, wherein the transform process definition includes a tree data structure.

57-58. (Canceled)

59. (Previously presented): A computer readable storage media having embodied thereon data, the data comprising:

computer instructions configured to position a definition pointer to point at a first compound transform definition, the first compound transform definition being within a transform process definition;

computer instructions configured to invoke a first parallel processing thread to read the pointed at first compound transform definition and sub-definitions of the first compound transform definition;

computer instructions configured to ~~increment~~ position a first payload pointer~~,~~ ~~within the data to be transformed.~~ to point at a data element to be transformed, the ~~incrementation~~ positioning being responsive to a data structure of the ~~pointed at~~ first compound transform definition;

computer instructions configured to call a dynamic function defined in the first compound transform definition, the dynamic function located elsewhere from the definition pointer position;

computer instructions configured to transform ~~any found~~ the data element into an output data file, responsive to the read first compound transform definition and called dynamic function;

computer instructions configured to ~~increment~~ position a second payload pointer~~,~~ ~~within the data to be transformed,~~ to point at a data element to be transformed, the ~~incrementation~~ positioning being responsive to a data structure of the ~~pointed at~~ second compound transform definition;

computer instructions configured to invoke a second parallel processing thread to read the pointed at second compound transform definition and sub-definitions of the second compound transform definition; and

computer instructions configured to transform the second data element into the output data file, responsive to the read second compound transform definition.

60. (Original): The computer readable media of claim 59, wherein the data further comprises computer instructions configured to employ recursion to transform a compound data element within the data to be transformed.

61-64. (Canceled)

65. (Currently amended): An application system comprising:

a computing device;

means for positioning a definition pointer to point at a first compound transform definition within a transform process definition;

means for invoking a first parallel processing thread to read the first compound transform definition by the computing device;

means for calling a dynamic function defined in the first compound transform definition, the dynamic function located elsewhere from the definition pointer position;

means for positioning the definition pointer to point at a second compound transform definition within the transform process definition;

means for invoking a second parallel processing thread to read the second compound transform definition by the computing device;

means for positioning a payload pointer to point to a first data element, the first data element being a member of a plurality of data elements within data to be transformed; and

means for generating an output data file using the first data element and the first and second compound transform definitions;

wherein the means for positioning the definition pointer and the means for positioning the payload pointer are enabled to be invoked concurrently.

66. (Original): The application system of claim 65, further including means for selecting the transform process definition from a set of transform process definitions, responsive to data associated with the data to be transformed.

67. (Original): The application system of claim 65, wherein a second data element has no contribution to output data generated using the transform process definition, the second data element being a member of the plurality of data elements.

68. (Previously presented): The application system of claim 65, further including means for adding data to the output data file, the added data being configured

responsive to the transform process definition and having no contribution from the data to be transformed.